1. Problématique :

L'algorithme doit prédire le si Tweet peut lui être proposé en fonction de son nombre de retweet sur le sujet.

2. Contrainte :

Seul le texte sera passé dans l'api et celui-ci sera proposé aux utilisateurs présents dans le data si la prédiction est supérieure à 30%

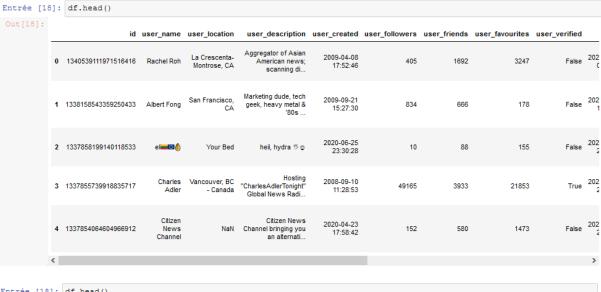
3. Présentation donnée

Dataset choisis est un dataset sur les retweets sur le nombre de retweet à propos du vaccin

Celui est composé de 2207 ligne x 16 colonnes.

Les colonnes que nous utiliserons pour la prédiction seront le nombres retweet ainsi que le tweet(text).

Forme du dataset (voir les captures ci-dessous)



Entrée [18]: df.head()											
Out[18]:	iser_followers	user_friends	user_favourites	user_verified	date	text	hashtags	source	retweets	favorites	is_retweet
	405	1692	3247	False	2020-12-20 06:06:44	Same folks said daikon paste could treat a cyt	['PfizerBioNTech']	Twitter for Android	0	0	False
	834	666	178	False	2020-12-13 16:27:13	While the world has been on the wrong side of 	NaN	Twitter Web App	1	1	False
	10	88	155	False	2020-12-12 20:33:45	#coronavirus #SputnikV #AstraZeneca #PfizerBio	['coronavirus', 'SputnikV', 'AstraZeneca', 'Pf	Twitter for Android	0	0	False
	49165	3933	21853	True	2020-12-12 20:23:59	Facts are immutable, Senator, even when you're	NaN	Twitter Web App	446	2129	False
	152	580	1473	False	2020-12-12 20:17:19	Explain to me again why we need a vaccine @Bor	['whereareallthesickpeople', 'PfizerBioNTech']	Twitter for iPhone	0	0	False
	<										>

4. Pipeline

A) DataHandler

Dans cette classe nous récuperons le dataset grâce a la méthode « get_data » ensuite nous faisons passer la dataframe dans un cleaner de text avec une regex ce que fais la méthode « clean_text ».

B) FeatureRecipe

Dans cette classe nous afficher séparons les données dans un tableau par types via la méthode « separate_variable_types » une fois cela fait nous supprimons les colonnes inutile avec la méthode « drop_uselessf » et ensuite nous allons supprimer les colonnes qui sont dupliqué via la méthode « drop_duplicate » une fois les colonnes inutile et dupliqué supprimer on vas vérifier si tout les data sont bien représenté si le nombre de NaN est supérieur a 3% on regroupe celle-ci dans une colonnes.

```
def __init__(esif, data pd.DetaFrame());
    self.inits = (1)
    self.inits = (1)
    self.init = (1
```

C)FeatureExtractor

Cette classe vas créer la pipeline soit le les data qui seront prise en compte et celle-ci qui seront testé pour la précision de l'algorithme

D)ModelBuild

Dans cette classe nous allons dans un premier temps entrainer celui-ci via la méthode « train » ensuite nous allons tester une prediction avec la méthode « predict_test » ensuite nous allons affiché la précision avec la méthode « print_accuracy » et enfin sauvegarder le model avec la méthode « save_model ».

```
class ModelBuild:
    def __init__(self, model_path, save, n_estimators):
          constructeur
          self.model_filename = model_path
          self.saveModel = save
self.date = date.today().isoformat()
          self.n_estimators=n_estimators
    def train(self,clf,X,Y):
    clf.fit(X, Y)
    def predict_test(self,clf,text):
          clf.predict_proba([text])[0]
    def print_accuracy(self,clf,X,Y):
          affichage de la precision des predictions
          accuracy=clf.score(X,Y)
          print('precision : {}'.format(accuracy))
    def FeatureImportance(self,X,Y,clf):
              attribut un score aux valeurs utilis? pour la prediction bas?
          sur leurs utilit?
         importance = clf.coef_()
for i,v in enumerate(importance):
    print('Feature: %0d, Score: %.5f' % (i,v))
pyplot.bar([x for x in range(len(importance))], importance)
          pyplot.show()
    def save_model(self,clf):
         model_filename = "model.joblib.z"
joblib.dump((clf), model_filename)
     def calculData(self,clf,X,Y,text):
         self.train(clf, X, Y)
self.predict_test(clf, text)
          self.print_accuracy(clf,X,Y)
#self.FeatureImportance(X,Y,clf)
self.save_model(clf)
```

CLOUD COMPUTING

5. Api

L'Api de l'algorithme sera déployée sur un docker et ensuite mise en ligne sur heroku il y aura un formulaire demandant le texte à prédire et il affichera la précision de la prédiction.

6. Conclusion

En Conclusion l'algorithme est considéré comme overfitting car la regex n'est pas assez précise dans la suppression des caractère et il n'y a pas assez de donnée d'entrainement pour avoir un prédiction assez précise et correct.